

A signature scheme from the finite field isomorphism problem

Jeffrey Hoffstein^{1*}, Joseph H. Silverman^{1*}, William Whyte², and Zhenfei Zhang²

¹ Brown University, Providence, USA
{jhoff, jhs}@math.brown.edu

² OnBoard Security, Wilmington, USA
{wwhyte, zzhang}@onboardsecurity.com

Abstract. In a recent paper the authors and their collaborators proposed a new hard problem, called the *finite field isomorphism problem*, and they used it to construct a fully homomorphic encryption scheme. In this paper, we investigate how one might build a digital signature scheme from this new problem. Intuitively, the hidden field isomorphism allows us to convert short vectors in the underlying lattice of one field into generic looking vectors in an isomorphic field.

Keywords: finite field isomorphism problem, lattice-based signature

1 Introduction

In [3], the authors and their collaborators presented a new hard problem, the *Finite Field Isomorphism Problem*. We briefly recall the problem here. Let q be a prime, let \mathbb{F}_q be the finite field with q elements, and let $\mathbf{f}(x) \in \mathbb{F}_q[x]$ and $\mathbf{F}(y) \in \mathbb{F}_q[y]$ be irreducible monic polynomials of degree n . Then

$$\mathbb{X} := \mathbb{F}_q[x]/(\mathbf{f}(x)) \quad \text{and} \quad \mathbb{Y} := \mathbb{F}_q[y]/(\mathbf{F}(y)) \quad (1)$$

are isomorphic fields with q^n elements. As usual, we identify elements of \mathbb{X} and \mathbb{Y} with vectors having integer coordinates between $-\frac{1}{2}q$ and $\frac{1}{2}q$, and we use this identification measure the size of field elements. It is then an experimental observation that, except for trivial cases, the isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$ does not respect the Archimedian property of size. Indeed, when \mathbf{f} and \mathbf{F} are distinct monic irreducible polynomials, it is observed that polynomials within a sphere of small radius in \mathbb{X} appear to be essentially uniformly distributed in \mathbb{Y} , with respect to both the L^∞ and the L^2 norms.

Definition 1. *Finite Field Isomorphism Problems (FFI):* Let $k \geq 1$ be an integer, let \mathbb{X} and \mathbb{Y} be as in (1), let ϕ be an isomorphism $\phi : \mathbb{X} \xrightarrow{\sim} \mathbb{Y}$, let

* Hoffstein and Silverman's research partially supported by NSF #1561709

$1 \leq \beta \leq \frac{1}{2}q$ be a parameter, and let $\mathbb{X}[\beta]$ denote the set of $\mathbf{a}(x) \in \mathbb{X}$ with L^∞ -norm bounded by $\|\mathbf{a}\| \leq \beta$. Choose $\mathbf{a}_1(x), \dots, \mathbf{a}_k(x)$ uniformly and randomly from $\mathbb{X}[\beta]$, and let $\mathbf{A}_i = \phi(\mathbf{a}_i)$ for $1 \leq i \leq k$ be the corresponding images in \mathbb{Y} .

The Computational FFI problem (CFFI): Given \mathbb{Y} and the list of polynomials $\mathbf{A}_1(y), \dots, \mathbf{A}_k(y)$, recover $\mathbf{f}(x)$ and/or one or more of $\mathbf{a}_1(x), \dots, \mathbf{a}_k(x)$.

The Decisional FFI problem (DFFI): Let $\epsilon > 0$. Let $\mathbf{b}_1(x)$ be randomly chosen in $\mathbb{X}[\beta]$, let $\mathbf{B}_1(y) = \phi(\mathbf{b}_1)$, and let $\mathbf{B}_2(y)$ be randomly chosen in \mathbb{Y} . Given the data $\mathbb{Y}, \mathbf{A}_1(y), \dots, \mathbf{A}_k(y)$, and given the pair $\{\mathbf{B}_1(y), \mathbf{B}_2(y)\}$ in a random order, identify, with probability greater than $1/2 + \epsilon$, which element of the pair was constructed using ϕ .

Remark 1. Under reasonable independence assumptions and for reasonable parameters, the CFFI has a unique solution. Thus for randomly chosen $\mathbf{F}(y)$ and $\mathbf{A}_1(y), \dots, \mathbf{A}_k(y)$, it is an exercise to estimate the probability that there exists an $\mathbf{f}(x) \in \mathbb{X}[\beta]$ and $\mathbf{a}_1(x), \dots, \mathbf{a}_k(x) \in \mathbb{X}[\beta]$ and an isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$ sending \mathbf{a}_i to \mathbf{A}_i for all $1 \leq i \leq k$. This probability is roughly $n^{-1}((2\beta + 1)^{k+1}/q^k)^n$.

In [3], the authors gave a detailed construction of how to build an isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$, described a preliminary analysis of the hardness of the FFI problems, and constructed a new fully homomorphic encryption scheme from the decisional version of the FFI problem. In this paper we explain how to build a signature scheme from the computational version of the FFI problem via the following framework, where we refer the reader to Section 3.3 for the definitions of the lattices $L_{\mathbf{h}}$ and $L_{\mathbf{H}}$.

1. Generate a signature \mathbf{s} , which is a short vector within or close to a lattice $L_{\mathbf{h}}$ related to the hidden field \mathbb{X} .
2. Publish its image $\mathbf{S} \in \mathbb{Y}$, and demonstrate the validity of the signature by showing a relationship between \mathbf{S} and a lattice $L_{\mathbf{H}}$ related to the public field \mathbb{Y} .

Since we have assumed that the map $\mathbb{X} \rightarrow \mathbb{Y}$ behaves like a random mapping, there is a negligible probability that the public lattice $L_{\mathbf{H}}$ will have any exceptionally short vectors. Therefore, we can build trapdoors using short vectors in $L_{\mathbf{h}}$ without the necessity of concealing the trapdoor from the attacker. This allows us to use very efficient methods to generate \mathbf{s} . As an example, we will show how this can be done using NTRU lattices

Verification is still possible due to the homomorphic property of the map $\mathbb{X} \rightarrow \mathbb{Y}$, but various lattice attacks on the public key, e.g., searching for the trapdoor from the lattice, are blunted or eliminated due to the non-linear nature of the isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$.

In this paper, we instantiate the above idea using the pqNTRUSign signature scheme [9,8,10], a candidate in the NIST PQC competition [17]. We name this new scheme pqFFSign.

We briefly recall that an NTRU lattice L is built using a ring of the form $\mathbb{X} := \mathbb{Z}[X]/\mathbf{f}(X)\mathbb{Z}[X]$, where $\mathbf{f}(X) = X^N \pm 1$ or some similar polynomial with

small coefficients [9]. More precisely, one creates a sublattice of $\mathbb{X} \times \mathbb{X}$ by choosing small secret polynomials and taking solutions to a congruence modulo a public integer q . The private key is the short lattice vectors coming from the small polynomials, and the public key is a basis consisting of long vectors. The secret short vectors, as usual, can be used to solve approximate closest vector problems, and a signature in the pqNTRUSign scheme consists of a solution to an approx-CVP, where an additional congruence modulo a small prime p is used to tie the document to the signature. Details are given in the cited references, but the important point is that the NTRU ring and the NTRU lattice are public values, and the NTRU lattice contains one or more very short vectors, a property that could potentially be exploited by lattice reduction algorithms.

The key idea in the present paper is that we do not allow the attacker to see the lattice \mathbb{X} , which contains one or more vectors that are considerably shorter than predicted by the Gaussian heuristic. Instead, we use the isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$ to transfer the entire problem to a lattice that does not have any especially short vectors. In this way, some previously described attacks against the private key of NTRU lattices, such as the hybrid attack [11], become impossible, since the very short vectors that exist in an NTRU lattice are mapped to random-looking vectors in the image lattice. However, even for pqFFSign, forgery attack via transcript analysis is still possible, so we rely as usual on rejection sampling [13,4] to seal the information leakage in transcripts. Due to space constraints, we omit a full description and analysis of rejection sampling here, but it is easy to adapt the material already described in the original pqNTRUSign paper.

Lattice-based signatures and rejection sampling. Signature schemes based on hard lattice problems have a history of almost 20 years. Early lattice-based signature schemes, such as GGHSig [6] and NTRUSign [7], leaked private key information via transcripts of message/signature pairs. An attacker could create a signing key from a long enough transcript using methods for “learning a parallelepiped” [5,16].

In [13], Lyubashevsky proposed a rejection sampling method to thwart transcript leakage attacks. Using his technique, signatures are produced according to a fixed public distribution, typically either a Gaussian or a uniform distribution. A transcript reveals only this public distribution, and contains no information about the particular signing key that is used to generate the signatures. This technique has become the de facto method for avoiding transcript leakage in lattice-based signature schemes; cf. as [4,14,8,10]. However, to ensure that the output signature follows a given distribution, a large number of randomly generated candidate signatures may need to be rejected before a suitable signature is accepted. This may significantly slow down the signing procedure.

2 Preliminaries

2.1 Notation

Let $\mathbf{f}(x) \in \mathbb{F}_q[x]$ and $\mathbf{F}(y) \in \mathbb{F}_q[y]$ be monic irreducible polynomials of degree n . We use \mathbf{f} and \mathbf{F} to construct two copies of \mathbb{F}_{q^n} , which we denote by

$$\mathbb{X} := \mathbb{F}_q[x]/(\mathbf{f}(x)) \quad \text{and} \quad \mathbb{Y} := \mathbb{F}_q[y]/(\mathbf{F}(y)),$$

and we let $\phi : \mathbb{X} \rightarrow \mathbb{Y}$ be an isomorphism of fields. In general, polynomials denoted by lower case letters will be polynomials in \mathbb{X} , and their isomorphic images in \mathbb{Y} will be denoted with the corresponding capital letters. A vector form of a polynomial is the vector consisting of all coefficients of the given polynomial. We will often identify polynomials and vectors when there is no ambiguity.

Consider a polynomial $\mathbf{a}(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \in \mathbb{X}$. We will informally say that $\mathbf{a}(x)$ is *short* if for all i , the congruence class $a_i \bmod q$ reduced into the interval $(-q/2, q/2]$ is small relative to q . An important class of such polynomials are those satisfying $a_i \in \{-1, 0, 1\}$; these are called *ternary polynomials*. We denote by

$$\|\mathbf{a}\| = \max_{0 \leq i \leq n-1} |a_i| \quad \text{and} \quad \|\mathbf{a}\|_2 = (a_0^2 + \cdots + a_{n-1}^2)^{1/2}$$

the L^∞ -norm and the L^2 -norm of \mathbf{a} , respectively, where it is understood that the coefficients of \mathbf{a} are normalized to lie in the interval $(-q/2, q/2]$. N.B. In our notation, the unsubscripted absolute value $\|\cdot\|$ is the L^∞ -norm, not the usual Euclidean L^2 -norm.

We now list some additional notation that is used in the rest of this paper:

- [Secret] $\mathbf{a}(x), \mathbf{b}(x) \in \mathbb{X}$ short irreducible monic polynomials of degree n .
- [Secret] $\mathbf{h}(x) \equiv \mathbf{b}(x) \cdot (\mathbf{p}\mathbf{a}(x))^{-1} \pmod{\mathbf{f}(x)} \in \mathbb{X}$.
- [Public] $\mathbf{H}(y) \in \mathbb{Y}$ is the image in \mathbb{Y} of $\mathbf{h}(x) \in \mathbb{X}$.
- [Secret] U is an n -by- n invertible matrix with small entries, e.g., in $\{-1, 0, 1\}$.

2.2 Two Uniformity Heuristics

We start with a heuristic which says that polynomials in $\mathbb{F}_q[x]$ with small coefficients are as likely to be irreducible as random polynomials.

Heuristic 1 *Let q be odd, and let $1 \leq \beta \leq \frac{1}{2}q$. Then there are approximately $\frac{1}{n} (\lfloor 2\beta + 1 \rfloor)^n$ distinct irreducible monic polynomials \mathbf{a} over $\mathbb{F}_q[x]$ satisfying $\|\mathbf{a}\| \leq \beta$.*

Heuristic 1 is based on the very reasonable assumption that monic irreducible polynomials are uniformly distributed over $\mathbb{F}_q[x]$ with respect to the L^∞ -norm, together with the well known prime number theorem for function fields, which states that the number of distinct irreducible monic polynomials of degree n in $\mathbb{F}_q[x]$ is on the order of q^n/n ; see [12, Chapter 7, Section 2, Corollary 2].

Similarly, classical primality tests for integers such as Miller–Rabin [15,18] are easily adapted to the function field setting. It is thus easy to check, at least with very high probability, whether a given polynomial is irreducible, and the probability of success is roughly $1/n$.

We invoke Heuristic 1 primarily to ensure that the signer will be able to find a pqFFSign private key. It also could help with combinatorial security in the sense that it says that the space of pqFFSign private keys is too large to search. However, since there does not appear to be an algorithm that directly lists the irreducible polynomials in the set of bounded coefficient polynomials, the actual combinatorial security comes from the size of the full set of bounded coefficient polynomials.

We will also mention the following additional uniformity heuristic on inverses, which might help in future security analyses of pqFFSign. However, we note that this heuristic is not related to the hardness of the finite field isomorphism problem.

Heuristic 2 *Let $q \geq 2$, and let $U \in \text{GL}_n(\mathbb{F}_q)$ be an invertible matrix with small entries, for example entries randomly chosen from $\{-1, 0, 1\}$. Then the entries of U^{-1} are approximately uniformly distributed in \mathbb{F}_q .*

This is similar to various well-established assumptions. Uniformity of products of the form $U_1^{-1}U_2$, with U_1 and U_2 small-entry circulant matrices, was used in analyzing the security of NTRUEncrypt [9]. If we instead choose the coefficients of U_1 and U_2 from a discrete Gaussian distribution for certain parameters, then it is proven in [19] that $U_1^{-1}U_2$ is almost uniformly distributed in $\text{GL}_n(\mathbb{F}_q)$.

We note that Heuristic 2 says that when our secret basis $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$ for \mathbb{X} is written as linear combinations of the (almost) standard basis x, x^2, \dots, x^n , the coefficients of those linear combinations look reasonably random in \mathbb{F}_q .

3 The pqFF-Sign Signature Scheme

For this section, we fix the following parameters:

- n , the degree of the polynomials $\mathbf{f}(x)$ and $\mathbf{F}(Y)$.
- q , a (moderate) size prime.
- β , a size parameter satisfying $1 \leq \beta \leq \frac{1}{2}q$, used to specify the size of the coefficients of a “small” polynomial.
- p , a (very small) prime different from q .
- B , a closeness parameter, used to determine if a claimed signature is a good enough solution to a CVP to be a valid signature.

3.1 An Algorithm to Find an Isomorphism

We recall how to find suitable polynomials $\mathbf{f}(x)$ and $\mathbf{F}(y)$ and an explicit isomorphism

$$\phi : \mathbb{F}_q[x]/(\mathbf{f}(x)) \xrightarrow{\sim} \mathbb{F}_q[y]/(\mathbf{F}(y))$$

as described in more detail in [3]. Recall that we need to find four polynomials $\mathbf{f}, \mathbf{F}, \phi, \psi$ satisfying:

- $\mathbf{f}(x) \in \mathbb{F}_q[x]$ is irreducible monic of degree n with $\|\mathbf{f}(x)\| \leq \beta$.
- $\mathbf{F}(y) \in \mathbb{F}_q[y]$ is irreducible monic of degree n with random coefficients.
- $\phi(y) \in \mathbb{F}_q[y]$ and $\psi(x) \in \mathbb{F}_q[x]$ have degree less than n .
- $\mathbf{F}(y) \mid \mathbf{f}(\phi(y))$.
- $\phi(\psi(x)) \equiv x \pmod{\mathbf{f}(x)}$.

The key idea here is that x will be identified with $\phi(y)$ and y will be identified with $\psi(x)$, and the conditions on ϕ and ψ say that this identification gives an explicit isomorphisms. The algorithm for finding these quantities is sketched in Algorithm 1.

Remark 2. For a given \mathbf{f} and \mathbf{F} , there are exactly n choices for ϕ , namely the n roots of \mathbf{f} in \mathbb{Y} . (The general theory of finite fields ensures that \mathbf{f} splits completely in $\mathbb{Y} \cong \mathbb{F}_{q^n}$.) Alternatively, given one value of $\phi(y)$, the complete set of possibilities for $\phi(y)$ are $\{\phi(y)^{q^i} \pmod{\mathbf{F}(y)} : 0 \leq i < n\}$. These are exactly the $\text{Gal}(\mathbb{Y}/\mathbb{F}_q)$ -conjugates of ϕ , where $\text{Gal}(\mathbb{Y}/\mathbb{F}_q) \cong \text{Gal}(\mathbb{F}_{q^n}/\mathbb{F}_q)$ is cyclic of order n , generated by the q -power Frobenius map.

Algorithm 1 Finite Field Isomorphism Generation (See [3] for details.)

- 1: Sample $\mathbf{f}(x)$ uniformly from the set of monic degree n polynomials $\mathbf{a}(x) \in \mathbb{F}_q[x]$ satisfying $\|\mathbf{a}\| \leq \beta$ until finding an \mathbf{f} that is irreducible.
 - 2: Sample $\mathbf{F}(y)$ uniformly from the set of monic degree n polynomials in $\mathbb{F}_q[y]$ until finding an \mathbf{F} that is irreducible. (Alternatively, use an \mathbf{F} provided by a trusted source.)
 - 3: Find a root of $\mathbf{f}(y)$ in the finite field $\mathbb{Y} := \mathbb{F}_q[y]/(\mathbf{F}(y)) \cong \mathbb{F}_{q^n}$, and lift this root to a polynomial $\phi(y) \in \mathbb{F}_q[y]$ of degree less than n .
 - 4: Construct the unique polynomial $\psi(x) \in \mathbb{F}_q[x]$ of degree less than n satisfying $\psi(\phi(y)) \equiv y \pmod{\mathbf{F}(y)}$.
 - 5: **return** $\mathbf{f}(x), \mathbf{F}(y), \phi(y)$ and $\psi(x)$.
-

3.2 The Detailed Scheme

The pqFFSign signature scheme uses three algorithms: KEYGEN, SIGNING and VERIFY. In addition it also requires a Hash function

$$\text{Hash} : \{\text{documents}\} \times \{\text{public keys}\} \longrightarrow \left(-\frac{1}{2^p}, \frac{1}{2^p} \right)^{2n}$$

that maps a document and a public key into a $2n$ -dimensional vector with small coefficients. We assume as usual that Hash is a cryptographically secure hash function in which each bit of the given document and each bit of the given public key affects every bit of the output.

Key generation The key generation algorithm

$$\text{KEYGEN}(\lambda) \rightarrow \mathbf{pk}, \mathbf{sk}$$

takes as the input a bit-security parameter λ , i.e., the goal is a scheme whose running time is $O(2^\lambda)$. It outputs a public key \mathbf{pk} and a secret key \mathbf{sk} as follows:

- Generate a parameter set $\Xi = \{n, p, q, \beta, B\}$ as a function of λ , where in particular p is a small integer, co-prime to q , satisfying $p^{n/2} \geq 2^\lambda$. (This ensures that a collision search on a set of size p^n is infeasible for the desired bit-security.)
- Generate a finite field isomorphism $\{\mathbf{f}, \mathbf{F}, \phi, \psi\}$ as described in Section 3.1.
- Generate polynomials $\mathbf{a}(x)$ and $\mathbf{b}(x)$ in $\mathbb{F}_q[x]$ with coefficients bounded by β .
- Compute $\mathbf{h}(x) \equiv (p\mathbf{a}(x))^{-1}\mathbf{b}(x) \pmod{\mathbf{f}(x)} \in \mathbb{X}$.³
- Compute $\mathbf{H}(y) := \mathbf{h}(\phi(y)) \in \mathbb{Y}$, the image of $\mathbf{h}(x)$ in \mathbb{Y} .
- Choose an invertible n -by- n matrix $U \in \text{GL}_n(\mathbb{F}_q)$ with small coefficients, e.g., with coefficients bounded by β .
- Define $\mathbf{c}_1(x), \mathbf{c}_2(x), \dots, \mathbf{c}_n(x) \in \mathbb{X}$ by the relation

$$U \begin{pmatrix} \mathbf{c}_1(x) \\ \mathbf{c}_2(x) \\ \vdots \\ \mathbf{c}_n(x) \end{pmatrix} \equiv \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^n \end{pmatrix} \pmod{q, \mathbf{f}(x)}. \quad (2)$$

(See Remark 5 for the significance of the relation (2).)

- Compute the images $\mathbf{C}_1(y), \dots, \mathbf{C}_n(y) \in \mathbb{Y}$ of $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$.
- Output the following signing key \mathbf{sk} and verification key \mathbf{pk} :

$$\begin{aligned} \mathbf{pk} &:= \{\Xi, \mathbf{F}(y), \mathbf{H}(y), \mathbf{C}_1(y), \dots, \mathbf{C}_n(y)\}, \\ \mathbf{sk} &:= \{\Xi, \mathbf{f}(x), \phi(y), \psi(x), U, \mathbf{a}(x), \mathbf{b}(x), \mathbf{c}_1(x), \dots, \mathbf{c}_n(x)\}. \end{aligned}$$

Signing The signing algorithm

$$\text{SIGN}(\mu, \mathbf{sk}) \rightarrow \sigma$$

takes a message μ and a secret key \mathbf{sk} as the input, and outputs a signature σ as follows:

- Hash the message and the public key to form a pair of n -dimensional mod p vectors,

$$\text{Hash}(\mu, \mathbf{pk}) = (\bar{\boldsymbol{\delta}}, \bar{\boldsymbol{\epsilon}}) := (\bar{\delta}_1, \dots, \bar{\delta}_n, \bar{\epsilon}_1, \dots, \bar{\epsilon}_n).$$

- Generate $(\boldsymbol{\delta}, \boldsymbol{\epsilon})$ satisfying:

$$\begin{aligned} \boldsymbol{\delta} &\equiv \bar{\boldsymbol{\delta}} \pmod{p}, & \|\boldsymbol{\delta}\| &\leq \frac{1}{2}q - B, \\ \boldsymbol{\epsilon} &\equiv \bar{\boldsymbol{\epsilon}} \pmod{p}, & \|\boldsymbol{\epsilon}\| &\leq \frac{1}{2}q - B, \end{aligned}$$

³ For comparison purposes, we note that $\mathbf{h}(x)$ has the form of a typical NTRU public key.

and with the property that the polynomials

$$\mathbf{s}(x) := \sum_{i=1}^n \delta_i \mathbf{c}_i(x) \quad \text{and} \quad \mathbf{t}(x) := \sum_{i=1}^n \epsilon_i \mathbf{c}_i(x), \quad (3)$$

satisfy the relation

$$\mathbf{s}(x)\mathbf{h}(x) \equiv \mathbf{t}(x) \pmod{q, \mathbf{f}(x)}$$

in the field \mathbb{X} . An algorithm to compute a valid pair $(\boldsymbol{\delta}, \boldsymbol{\epsilon})$ is given in Section 3.3.

– **Output:** The signature is the pair $\sigma := (\boldsymbol{\delta}, \boldsymbol{\epsilon})$.

Verification The verification algorithm

$$\text{VERIFY}(\mu, \sigma, \mathbf{pk}) \rightarrow \text{ACCEPT/REJECT}$$

takes a message μ , a signature σ , and a public key \mathbf{pk} as the input. It first uses the message and the public key to compute

$$(\bar{\boldsymbol{\delta}}, \bar{\boldsymbol{\epsilon}}) := \text{Hash}(\mu, \mathbf{pk}).$$

It then checks the validity of the following conditions:

$$\boldsymbol{\delta} \equiv \bar{\boldsymbol{\delta}} \pmod{p}, \quad \|\boldsymbol{\delta}\| \leq \frac{1}{2}q - B, \quad (4)$$

$$\boldsymbol{\epsilon} \equiv \bar{\boldsymbol{\epsilon}} \pmod{p}, \quad \|\boldsymbol{\epsilon}\| \leq \frac{1}{2}q - B. \quad (5)$$

$$\left(\sum_{i=1}^n \delta_i \mathbf{C}_i(y) \right) \mathbf{H}(y) = \sum_{i=1}^n \epsilon_i \mathbf{C}_i(y) \quad \text{in } \mathbb{Y}. \quad (6)$$

(See Remark 3 for the purpose of these conditions.) **Output:** ACCEPT if (4), (5), and 6 are true, REJECT otherwise.

Remark 3. The mod p conditions in (4) and (5) for $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ serve to link the signature to the document and to the public key. The equality (6) and the norm conditions in (4) and (5) give a relation in \mathbb{Y} that reflects a relation among short vectors in \mathbb{X} .

Remark 4. A comparison shows that the primary difference between pqNTRU-Sign and pqFFSign is that in the former, signatures are created by using polynomials in the ring $\mathbb{Z}[x]/(\mathbf{f}(x))$ whose coefficients are small relative to the standard basis $1, x, \dots, x^{n-1}$, while in the latter we use polynomials with small coefficients relative to the basis $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$. The advantage of this approach is that the verifier only sees a relation involving $\mathbf{C}_1(y), \dots, \mathbf{C}_n(y)$ in \mathbb{Y} .

Remark 5. The polynomials $\mathbf{c}_1(x), \mathbf{c}_2(x), \dots, \mathbf{c}_n(x)$ form an \mathbb{F}_q -basis for \mathbb{X} , and $\mathbf{C}_1(y), \mathbf{C}_2(y), \dots, \mathbf{C}_n(y)$ form an \mathbb{F}_q -basis for \mathbb{Y} . Each $\mathbf{C}_j(y)$ is the image of the corresponding $\mathbf{c}_j(x)$ under the isomorphism that sends $x \mapsto \phi(y)$. This same isomorphism preserves the coefficients of linear combinations of the $\mathbf{c}_j(x)$, that is,

$$\sum \alpha_j \mathbf{c}_j(x) \mapsto \sum \alpha_j \mathbf{C}_j(y).$$

The key property on which pqFFSign is based is the relation (2) and the fact that the coefficients of U are small, from which it follows that each of the monomials x, x^2, \dots, x^n is a linear combination of the $\mathbf{c}_j(x)$ with *small coefficients*. From this it follows that any polynomial in x with small coefficients can in turn be written as a polynomial in $\mathbf{c}_j(x)$ with small coefficients. Note that Heuristic 2 says that the converse will not be true, i.e., it says that the coefficients of the $\mathbf{c}_j(x)$ will be uniformly distributed mod q .

3.3 Algorithm To Find (δ, ϵ)

We note that the choice of the security parameter B provides a balance between combinatorial security (large B is good) and the difficulty of generating a valid signature using the private key (small B is good).

Definition 2. For any polynomial $\mathbf{h}(x) \in \mathbb{X} = \mathbb{F}_q[x]/(\mathbf{f}(x))$, we define the associated NTRU lattice to be the $2n$ -dimensional lattice

$$L_{\mathbf{h}} := \left\{ (\mathbf{u}(x), \mathbf{v}(x)) \in \mathbb{Z}[x]^2 : \begin{array}{l} \deg(\mathbf{u}) \leq n-1, \deg(\mathbf{v}) \leq n-1, \\ \mathbf{v}(x) \equiv \mathbf{u}(x)\mathbf{h}(x) \pmod{q, \mathbf{f}(x)} \end{array} \right\}.$$

Similarly for $L_{\mathbf{H}}$. We note that $(\mathbf{p}\mathbf{a}(x), \mathbf{b}(x))$ is a short vector in $L_{\mathbf{h}}$, but its image in $L_{\mathbf{H}}$ is unlikely to be short.

1. For $j = 1, \dots, n$, choose $\delta_j^{(0)}$ at random such that $|\delta_j^{(0)}| < \frac{1}{2}q - B$ and $\delta_j^{(0)} \equiv \bar{\delta}_j \pmod{p}$ and set

$$\mathbf{s}_0(x) = \sum_{j=1}^n \delta_j^{(0)} \mathbf{c}_j(x).$$

2. Define $\mathbf{t}_0(x)$ by

$$\mathbf{t}_0(x) \equiv \mathbf{s}_0(x)\mathbf{h}(x) \pmod{q, \mathbf{f}(x)}$$

and write

$$\mathbf{t}_0(x) \equiv \sum_{i=0}^{n-1} t_i x^i \pmod{q, \mathbf{f}(x)}.$$

Note that by construction we have $(\mathbf{s}_0(x), \mathbf{t}_0(x)) \in L_{\mathbf{h}}$.

3. Rewrite $\mathbf{t}_0(x)$ as a linear combination of the basis $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$ of \mathbb{X} , say

$$\mathbf{t}_0(x) = \sum_{j=1}^n \eta_j \mathbf{c}_j(x) \quad \text{for some } \eta_1, \dots, \eta_n \text{ with } -\frac{1}{2}q < \eta_j \leq \frac{1}{2}q.$$

If all of the η_j lie in the interval $(-\frac{1}{2}q + B, \frac{1}{2}q - B]$, proceed to Step (iv); otherwise go back to Step (i) and choose new values for the $\delta_j^{(0)}$. (For appropriate choices of parameters, the probability of success at this step is reasonably high; see [8] for details. As described in [8], this step may be used to implement rejection sampling, which provides security against transcript attacks.)

4. Construct $(\mathbf{u}(x), \mathbf{v}(x)) \in L_{\mathbf{h}}$ such that

$$\mathbf{u}(x) = \sum_{j=1}^n \delta_j^{(u)} \mathbf{c}_j(x) \quad \text{and} \quad \mathbf{v}(x) = \sum_{j=1}^n \delta_j^{(v)} \mathbf{c}_j(x),$$

with $|\delta_j^{(u)}|, |\delta_j^{(v)}| < B$ for all j , with $\delta_j^{(u)} \equiv 0 \pmod{p}$, and $\delta_j^{(v)} + \eta_j \equiv \bar{\epsilon}_j \pmod{p}$ for all j . The procedure for this step is sufficiently complicated that we give the details in Section 3.3.1.

5. Set $\mathbf{s}(x) = \mathbf{s}_0(x) + \mathbf{u}(x)$ and $\mathbf{t}(x) = \mathbf{t}_0(x) + \mathbf{v}(x)$. Write $\mathbf{s}(x)$ and $\mathbf{t}(x)$ as linear combination of the basis polynomials $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$ as in (3), and read off the coefficients of those linear combinations to create the vectors $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ that form the signature.

3.3.1 Details of Step 4 To construct the desired $(\mathbf{u}(x), \mathbf{v}(x))$, we construct an appropriate $\mathbf{r}(x)$ which is short, and set

$$\mathbf{u}(x) = p\mathbf{r}(x)\mathbf{a}(x) \quad \text{and} \quad \mathbf{v}(x) = \mathbf{r}(x)\mathbf{b}(x).$$

We want to find an $\mathbf{r}(x)$ that satisfies

$$\mathbf{r}(x)\mathbf{b}(x) = \sum_{j=1}^n \delta_j^{(v)} \mathbf{c}_j(x) \quad \text{with the } |\delta_j^{(v)}| < B \text{ and } \delta_j^{(v)} + \eta_j \equiv \bar{\epsilon}_j \pmod{p},$$

and also satisfies

$$p\mathbf{r}(x)\mathbf{a}(x) = \sum_{j=1}^n \delta_j^{(u)} \mathbf{c}_j(x) \quad \text{with the } |\delta_j^{(u)}| < B \text{ and } \delta_j^{(u)} \equiv 0 \pmod{p}.$$

Suppose first that we have any $\mathbf{r}(x)$ that is sufficiently short. Since $\mathbf{a}(x)$ is short, we see that $\mathbf{r}(x)\mathbf{a}(x)$ is also short, and we may write

$$\mathbf{r}(x)\mathbf{a}(x) = \sum_{i=1}^n d_i x^i \in \mathbb{X} \quad \text{with small } d_i.$$

Then the $\delta_j^{(u)}$ s in the formula

$$p\mathbf{r}(x)\mathbf{a}(x) = \sum_{j=1}^n \delta_j^{(u)} \mathbf{c}_j(x)$$

are given by

$$(\delta_1^{(u)}, \dots, \delta_n^{(u)}) = p(d_1, \dots, d_n)U.$$

As long as all of the d_i and all of the entries of U are sufficiently small, we will have $|\delta_j^{(u)}| < B$ and $\delta_j^{(u)} \equiv 0 \pmod{p}$ for all j . Thus for whatever sufficiently short $\mathbf{r}(x)$ we choose, the $\delta_j^{(u)} \equiv 0 \pmod{p}$ condition will hold.

We turn now to finding a short $\mathbf{r}(x)$ that satisfies

$$\mathbf{r}(x)\mathbf{b}(x) = \sum_{j=1}^n \delta_j^{(v)} \mathbf{c}_j(x) \quad \text{with } |\delta_j^{(v)}| < B \text{ and } \delta_j^{(v)} \equiv \bar{\epsilon}_j - \eta_j \pmod{p}.$$

To accomplish this, write $\mathbf{b}(x) = \sum_{i=0}^{n-1} b_i x^i$, set

$$(b_{0,0}, b_{0,1}, \dots, b_{0,n-1}) = (b_0, b_1, \dots, b_{n-1}),$$

and for $1 \leq i < n$, define $(b_{i,0}, b_{i,1}, \dots, b_{i,n-1})$ by

$$x^i \mathbf{b}(x) \equiv b_{i,0} + b_{i,1}x + \dots + b_{i,n-1}x^{n-1} \pmod{\mathbf{f}(x)}.$$

Let \mathcal{B} denote the matrix whose i, j entry is $b_{i,j}$ and consider the product $\mathcal{B}U$. The entries of $\mathcal{B}U$ are small because the $b_{i,j}$ and the entries of U are small. For any

$$\mathbf{r}(x) = \sum_{i=0}^{n-1} r_i x^i \equiv \sum_{i=1}^n r'_i x^i \pmod{\mathbf{f}(x)}$$

we have

$$\mathbf{r}(x)\mathbf{b}(x) = (r'_1, r'_2, \dots, r'_n) \mathcal{B}U \begin{pmatrix} \mathbf{c}_1(x) \\ \mathbf{c}_2(x) \\ \vdots \\ \mathbf{c}_n(x) \end{pmatrix}.$$

To solve for $\mathbf{r}(x)$, first define

$$(\bar{r}'_1, \bar{r}'_2, \dots, \bar{r}'_n) \equiv (\bar{\delta}_1^{(v)}, \bar{\delta}_2^{(v)}, \dots, \bar{\delta}_n^{(v)}) (\mathcal{B}U)^{-1} \pmod{p}$$

and lift each \bar{r}'_j to $r'_j \in (-p/2, p/2]$. Then define $\delta_j^{(v)}$ by

$$(\delta_1^{(v)}, \dots, \delta_n^{(v)}) = (r'_1, \dots, r'_n) \mathcal{B}U.$$

This accomplishes the goal

$$\mathbf{r}(x)\mathbf{b}(x) = \sum_{j=1}^n \delta_j^{(v)} \mathbf{c}_j(x) \quad \text{with } \delta_j^{(v)} \equiv \bar{\delta}_j^{(v)} \pmod{p}.$$

4 Security Considerations

We highlight some of best known attacks. Due to page limitations, we leave other (less effective) known attacks in the appendix.

4.1 The Size of B

The key point is to choose B in such a way that the final signature lies inside an appropriate subset of the $(-\frac{1}{2}q, \frac{1}{2}q]$ box. Recall that

$$(\delta_1^{(v)}, \dots, \delta_n^{(v)}) = (r'_1, \dots, r'_n) \mathcal{B}U.$$

The coefficients r'_i lie in the interval $(-p/2, p/2]$. Let K be chosen to be the maximum of the absolute values of the entries of $\mathcal{B}U$. Then each $|\delta_j^{(v)}|$ will be bounded above by a constant multiple of $pK\sqrt{n}$. The same almost applies to $|\delta_j^{(u)}|$, but because of the multiple of p it will be larger by a factor of p unless some scaling is done to compensate for this, for example, by choosing the original $\delta_j^{(0)}$ from an interval smaller than $(-q/2, q/2]$. So B must be on the order of $pK\sqrt{n}$. The size of K will be optimal when U and \mathcal{B} are as sparse as possible.

4.2 Recovering the Isomorphism/Solving CFFI Problem

The attacker is given polynomials $C_1(y), \dots, C_n(y) \in \mathbb{Y}$ that are the images of unknown short polynomials $c_1(x), \dots, c_n(x) \in \mathbb{X}$ via an unknown isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$. For the general CFFI problem, if the attacker knows at least $2n$ elements of \mathbb{Y} that are images of short elements of \mathbb{X} , then she can set up a mixed lattice/combinatorial attack to recover the short elements of \mathbb{X} and the isomorphism $\mathbb{X} \rightarrow \mathbb{Y}$. See Section B in the appendix for details.

This attack requires $2n$ elements, but the public key for pqFFSign provides the attacker with only n images of short elements of \mathbb{X} , not $2n$. So the attack would seem to fail at this point. However, the fact that $f(x)$ is small means that products of small elements of \mathbb{X} remain reasonably small. Indeed, that is a key fact being exploited by pqFFSign. So for $1 \leq i \leq n$, the attacker can create additional elements by taking products such as $C_{n+i}(y) := C_1(y)C_i(y) \bmod F(y)$ in \mathbb{Y} , and these new elements of \mathbb{Y} will be images of somewhat small elements of \mathbb{X} . This may allow the attack described in Section B to proceed, with the caveat that the target vectors will now be considerably larger than in the basic version of the CFFI problem. On the other hand, since the coordinates of the target vectors will now consist of n very small numbers and n moderately small numbers, the targets are unbalanced. So a full analysis of the underlying lattice problem requires balancing the lattice to account for this imbalance in the targets' coordinates.

4.3 Recovering the Unique Shortest Vector

There are two main security concerns that determine parameters in pqNTRUSign. One is the problem of recovering the private key from the public NTRU key, and the other is the problem of forgery. Of these, the one that has the biggest impact on parameter size is the public key to private key problem. This is because, to make rejection sampling efficient, the q needs to be chosen large compared to n .

This makes the lattice problem somewhat easier and forces an increase in the size of n . The forgery problem requires smaller parameters to achieve the same security levels.

In this context there appear at first to be two NTRU-type problems: Recovering $\mathbf{a}(x), \mathbf{b}(x)$ from $\mathbf{h}(x)$, and recovering the corresponding polynomials $\mathbf{A}(y), \mathbf{B}(y)$ from $\mathbf{H}(y)$. However, the polynomial $\mathbf{h}(x)$ is private and is only revealed if the underlying isomorphism is discovered, in which case the scheme is considered broken. So this lattice problem does not apply to pqFFSign.

On the other hand, the polynomial $\mathbf{H}(y)$ is public, but the corresponding problem of recovering $\mathbf{A}(y), \mathbf{B}(y)$ from $\mathbf{H}(y)$ is not a lattice problem because $\mathbf{A}(y)$ and $\mathbf{B}(y)$ are polynomials with coefficients that are essentially random mod q , so they are not short. This is a consequence of the fundamental observation that the isomorphism between the two copies of \mathbb{F}_q^n does not respect the archimedean properties of the polynomials' coefficients. Further, since $\mathbf{A}(y)$ and $\mathbf{B}(y)$ are not short, recovery of them does not appear to be helpful to the attacker.

There is a lattice attack to recover the matrix U from the $\mathbf{C}_j(y)$, and this would suffice to break the scheme, but the dimension of the lattice required to solve this problem is at least n^2 . We describe this attack in Section C. For all of these reasons, it thus appears that it suffices to set parameters to avoid forgery attacks, and this should allow for smaller signatures and better operating characteristics. In particular, by choosing the small prime p as close as possible to \sqrt{q} , we can make the Gaussian defect of a solution very close to one, which thus makes lattice reduction attacks very difficult even in relatively small dimensions.

5 Conclusion and Future Work

In this work we present a signature scheme (partially) based on the Computational Finite Field Isomorphism Problem (CFFI). Future research directions include:

The hardness of the finite field isomorphism problem: In this paper, we have indicated several ways in which one might try to solve the CFFI problem. However, the quantitative difficulty of the CFFI problem is presently unclear.

Average-case/worse-case hardness: There exists an easy instance of the CFFI problem, namely when $\mathbf{f}(x) = \mathbf{F}(y)$. It would be of interest to prove that random (or even, all) instances of the CFFI problem with $\mathbf{f}(x) \neq \mathbf{F}(y)$ are equally difficult.

Transcript security and rejection sampling: A sufficiently long raw pqNTRU-Sign transcript allows an attacker to reconstruct a short lattice basis due to the way in which signatures are generated. The use of rejection sampling eliminates this attack by leading to transcripts that are independent of the underlying lattice. (See [8] for details.) Similarly, raw pqFFSign reveals a transcript of short

vectors (δ, ϵ) that may contain information about \mathbf{f} or (\mathbf{a}, \mathbf{b}) or U . We expect that rejection sampling can be used to produce key-independent transcripts. Formulating and proving such a result should not be hard, but remains to be done.

Security reduction between pqFFSign and CFFI: It is clear that the security of pqFFSign, the signature scheme that we have proposed in this paper, relies on the difficulty of CFFI. The converse is not clear. It would be quite interesting to give a security reduction showing, say, that breaking pqFFSign plus an algorithm solving some sort of standard hard CVP lattice problem is equivalent to solving the CFFI problem.

Analyze the balanced lattice attack : As discussed in Section 4.2, the lattice attack (Section B) on the pure CFFI problem needs to be balanced before being applied to pqFFSign. Doing this will yield constraints on the various parameters required to achieve a desired level of security. Further, even if the lattice attack succeeds completely, there is still what appears to be a difficult combinatorial problem to solve. This combinatorial problem deserves further study.

References

1. Wieb Bosma, John Cannon and Catherine Playoust, The Magma algebra system. I. The user language., *J. Symbolic Comput.* **24** (1997), 235–265.
2. Yuanmi Chen and Phong Q. Nguyen, BKZ 2.0: Better Lattice Security Estimates, in: *ASIACRYPT*, pp. 1–20, 2011.
3. Yarkin Doröz, Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, Berk Sunar, William Whyte and Zhenfei Zhang, Fully Homomorphic Encryption from the Finite Field Isomorphism Problem, *PKC 2018* (2018).
4. Léo Ducas, Alain Durmus, Tancrede Lepoint and Vadim Lyubashevsky, *Lattice Signatures and Bimodal Gaussians*, CRYPTO 2013 (Ran Canetti and Juan A. Garay, eds.), LNCS 8042, Springer, 2013, pp. 40–56.
5. Léo Ducas and Phong Q. Nguyen, Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures, in: *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pp. 433–450, 2012.
6. Oded Goldreich, Shafi Goldwasser and Shai Halevi, Public-Key Cryptosystems from Lattice Reduction Problems, in: *CRYPTO*, pp. 112–131, 1997.
7. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman and William Whyte, NTRUSIGN: Digital Signatures Using the NTRU Lattice, in: *Topics in Cryptology - CT-RSA 2003, The Cryptographers’ Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, pp. 122–140, 2003.
8. Jeffrey Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman and William Whyte, Transcript Secure Signatures Based on Modular Lattices, in: *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, pp. 142–159, 2014.

9. Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman, NTRU: A Ring-Based Public Key Cryptosystem, in: *ANTS*, pp. 267–288, 1998.
10. Jeffrey Hoffstein, Jill Pipher, William Whyte and Zhenfei Zhang, *A signature scheme from Learning with Truncation*, Cryptology ePrint Archive, Report 2017/995, 2017, <http://eprint.iacr.org/2017/995>.
11. Nick Howgrave-Graham, A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU, in: *CRYPTO*, pp. 150–169, 2007.
12. K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, Springer-Verlag, 1990.
13. Vadim Lyubashevsky, *Fiat-shamir with aborts: Applications to lattice and factoring-based signatures*, ASIACRYPT 2009, Springer, 2009, pp. 598–616.
14. Vadim Lyubashevsky, *Lattice Signatures without Trapdoors*, EUROCRYPT 2012 (David Pointcheval and Thomas Johansson, eds.), LNCS 7237, Springer, 2012, pp. 738–755.
15. Gary L. Miller, Riemann’s hypothesis and tests for primality, *Journal of Computer and System Sciences* **13** (1976), 300 – 317.
16. Phong Q. Nguyen and Oded Regev, Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures, *J. Cryptology* **22** (2009), 139–160.
17. NIST, *Post-Quantum Cryptography - Round 1 Submissions*, <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions>.
18. Michael O Rabin, Probabilistic algorithm for testing primality, *Journal of Number Theory* **12** (1980), 128 – 138.
19. Damien Stehlé and Ron Steinfeld, Making NTRU as Secure as Worst-Case Problems over Ideal Lattices, in: *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pp. 27–47, 2011.

A Security Against Forgery

Recall that a signature is a pair of vectors $\delta = (\delta_1, \dots, \delta_n)$, and $\epsilon = (\epsilon_1, \dots, \epsilon_n)$, where $(\delta, \epsilon) \bmod p = (\bar{\delta}, \bar{\epsilon})$. Hence (δ, ϵ) is a vector in a translated lattice

$$(\delta, \epsilon) \in L_1 := p\mathbb{Z}^{2n} + (\bar{\delta}, \bar{\epsilon}).$$

Next, using the formulas

$$S(y) = \sum_{i=1}^n \delta_i C_i(y), \quad T(y) = \sum_{i=1}^n \epsilon_i C_i(y), \quad S(y)H(y) = T(y),$$

we have

$$\delta C(y)H(y) = \epsilon C(y).$$

That is, (δ, ϵ) is also in the lattice

$$(\delta, \epsilon) \in L_2 := \text{RowSpan} \begin{pmatrix} qI_n & 0 \\ CHC^{-1} & I_n \end{pmatrix}.$$

We observe that any vector in the intersection $L_1 \cap L_2$ is a potential signature. We have

$$\det(L_1) = p^{2n}, \quad \det(L_2) = q^n, \quad \det(L_1 \cap L_2) = p^{2n}q^n,$$

where the last equality uses the assumption that $\gcd(p, q) = 1$. Hence the Gaussian heuristic for the shortest nonzero vector in $L_1 \cap L_2$ is

$$\text{GH}(L_1 \cap L_2) = \sqrt{\frac{2 \dim(L_1 \cap L_2)}{\pi e}} \det(L_1 \cap L_2)^{\frac{1}{\dim}} = \sqrt{\frac{p^2 q N}{\pi e}}.$$

Further, we see that the L^2 norm of the target vector is bounded by

$$\|(\boldsymbol{\delta}, \boldsymbol{\epsilon})\|_2 \leq \sqrt{2Nq}.$$

This yields the root Hermite factor

$$\gamma(L_1 \cap L_2) = \left(\frac{\sqrt{2Nq}}{\sqrt{p^2 q N / \pi e}} \right)^{1/2n} = \left(\sqrt{2\pi e q / p} \right)^{1/2n}.$$

Hence a lattice attack will be infeasible if we choose parameters to ensure that

$$\left(\sqrt{2\pi e q / p} \right)^{1/2n} < \gamma_{\text{exp}},$$

where γ_{exp} is chosen to be the experimental Hermite factor expected to be achievable via lattice reduction algorithms. For example, using the LLL-BKZ 2.0 algorithms, a Hermite factor $\gamma = 1.005$ seems to be secure; see [2].

B A Combined Lattice-Combinatorial Attack on the CFFI Problem

We consider the CFFI problem under the assumption that $k \geq 2n$, so the attacker is given polynomials $\mathbf{A}_1(y), \dots, \mathbf{A}_k(y) \in \mathbb{Y}$ that are the images of polynomials $\mathbf{a}_1(x), \dots, \mathbf{a}_k(x) \in \mathbb{X}$ having small coefficients. We identify a polynomial $\mathbf{C}(y) = C_0 + \dots + C_{n-1}y^{n-1} \in \mathbb{Y}$ with the row vector $\mathbf{C} := (C_0, \dots, C_{n-1}) \in \mathbb{F}_q^n$, and similarly for elements of \mathbb{X} . Then the (unknown) isomorphism $\psi : \mathbb{Y} \rightarrow \mathbb{X}$ is given by an $n \times n$ matrix Ψ , i.e., Ψ is the matrix of ψ relative to the bases $\{1, \dots, y^{n-1}\}$ of \mathbb{Y} and $\{1, \dots, x^{n-1}\}$ of \mathbb{X} . Note that with this notation, we have $\psi(\mathbf{C}(y)) = \mathbf{C}\Psi$.

We form four $n \times n$ matrices

$$\mathfrak{M} := \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix}, \quad \mathfrak{N} := \begin{pmatrix} \mathbf{A}_{n+1} \\ \vdots \\ \mathbf{A}_{2n} \end{pmatrix}, \quad \mathbf{m} := \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{pmatrix}, \quad \mathbf{n} := \begin{pmatrix} \mathbf{a}_{n+1} \\ \vdots \\ \mathbf{a}_{2n} \end{pmatrix}.$$

In other words, the rows of \mathfrak{M} are the vectors $\mathbf{A}_1, \dots, \mathbf{A}_n$, and similarly for the other matrices. Then by assumption we have the matrix formulas

$$\mathfrak{M}\Psi = \mathbf{m} \quad \text{and} \quad \mathfrak{N}\Psi = \mathbf{n}, \quad \text{and thus} \quad \mathfrak{N}\mathfrak{M}^{-1}\mathbf{m} = \mathbf{n},$$

where all computations are done in \mathbb{F}_q . Note that although the matrix Ψ has unknown random coefficients, the matrices \mathbf{m} and \mathbf{n} have unknown *small* coefficients, and the matrices \mathfrak{M} and \mathfrak{N} are known. So this last formula will allow us to describe a known lattice with unknown short target vectors.

For each $1 \leq t \leq n$, let

$$\mathbf{v}_t := \begin{pmatrix} \text{column vector whose first } n \text{ coordinates are the } t^{\text{th}}\text{-column} \\ \text{of } \mathbf{m} \text{ and whose second } n \text{ coordinates are the } t^{\text{th}}\text{-column of } \mathbf{n}. \end{pmatrix} \in \mathbb{Z}^{2n},$$

where we have, as usual, lifted numbers from \mathbb{F}_q to an interval centered at 0. Then the relation $\mathfrak{N}\mathfrak{M}^{-1}\mathbf{m} = \mathbf{n}$ tells us that

$$\mathbf{v}_t \in \mathcal{L} := \text{ColumnSpan} \begin{pmatrix} I & 0 \\ \mathfrak{N}\mathfrak{M}^{-1} & qI \end{pmatrix}.$$

We may thus use lattice reduction methods to search for the n short vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ in the known lattice \mathcal{L} .

The lattice \mathcal{L} satisfies

$$\dim \mathcal{L} = 2n \quad \text{and} \quad \text{Det } \mathcal{L} = q^n,$$

so the Gaussian expected norm of the smallest vector in \mathcal{L} is

$$\text{GH}(\mathcal{L}) = \sqrt{\dim \mathcal{L} / \pi e} (\text{Det } \mathcal{L})^{1/\dim \mathcal{L}} = \sqrt{\frac{2nq}{\pi e}}.$$

On the other hand, if we assume that the coefficients of $\mathbf{a}_1, \dots, \mathbf{a}_{2n}$ are random integers in the interval from $-\beta$ to β , as in the statement of CFFI, then the expected length of each target vector \mathbf{v}_t is roughly $\tau := \sqrt{2n/3}\beta$. Hence the root Hermite ratio is

$$\gamma(\mathcal{L}) := \left(\frac{\text{GH}(\mathcal{L})}{\tau} \right)^{1/\dim \mathcal{L}} \approx \left(\frac{1}{\beta} \sqrt{\frac{3q}{\pi e}} \right)^{1/2n}.$$

We remark that currently a Hermite ratio smaller than 1.005 appears to achieve reasonable security; cf. [2].

There are two additional issues. First, it is not clear that the set

$$\mathcal{S} := \{\mathbf{v}_t : 1 \leq t \leq n\}$$

of target vectors consists of the n shortest linearly independent vectors in \mathcal{L} . Thus even a complete lattice reduction that finds an “optimal basis” for \mathcal{L} may only return some small linear combinations of the elements of \mathcal{S} . The attacker would then have to unsort these small linear combinations to find the set \mathcal{S} .

Second, even if the attacker finds the exact set of short vector \mathcal{S} , there is a combinatorial problem to solve, since the set \mathcal{S} comes with no preferred order. But the coordinates of \mathbf{v}_t are the coefficients of x^t in $\mathbf{a}_1, \dots, \mathbf{a}_{2n}$, so recovery of $\mathbf{a}_1, \dots, \mathbf{a}_{2n}$ and reconstruction of Ψ works only if the set of short vectors \mathcal{S} is put in the correct order. There are $n!$ ways to reorder \mathcal{S} , so if n is large, a full search, or even a collision search, is infeasible. It remains an open problem to find a faster method to correctly order the elements of \mathcal{S} .

C Attack to Recover U

During key generation we construct a set of polynomials $\mathbf{c}_1(x), \dots, \mathbf{c}_n(x)$ by choosing an $n \times n$ sparse matrix U and setting

$$\begin{pmatrix} \mathbf{c}_1(x) \\ \mathbf{c}_2(x) \\ \vdots \\ \mathbf{c}_n(x) \end{pmatrix} \equiv U^{-1} \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^n \end{pmatrix} \pmod{q}.$$

Then for example we have

$$\sum_{i=1}^n u_{1,i} \mathbf{c}_i(x) = x, \quad \sum_{i=1}^n u_{2,i} \mathbf{c}_i(x) = x^2, \quad \sum_{i=1}^n u_{3,i} \mathbf{c}_i(x) = x^3,$$

so using the fact that $x \cdot x^2 = x^3$, we find that

$$\left(\sum_{i=1}^n u_{1,i} \mathbf{c}_i(x) \right) \left(\sum_{j=1}^n u_{2,j} \mathbf{c}_j(x) \right) \equiv \left(\sum_{k=1}^n u_{3,k} \mathbf{c}_k(x) \right) \pmod{f(x), q}.$$

This in turn gives a formula in \mathbb{Y} ,

$$\left(\sum_{i=1}^n u_{1,i} \mathbf{C}_i(y) \right) \left(\sum_{j=1}^n u_{2,j} \mathbf{C}_j(y) \right) \equiv \left(\sum_{k=1}^n u_{3,k} \mathbf{C}_k(y) \right) \pmod{F(y), q}.$$

Multiplying this out gives

$$\begin{aligned} \sum_{i=1}^n \underbrace{u_{1,i} u_{2,i}}_{v_i} \mathbf{C}_i(y)^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \underbrace{(u_{1,i} u_{2,j} + u_{1,j} u_{2,i})}_{w_{ij}} \mathbf{C}_i(y) \mathbf{C}_j(y) \\ \equiv \left(\sum_{k=1}^n u_{3,k} \mathbf{C}_k(y) \right) \pmod{F(y), q}. \end{aligned}$$

Reducing the various $\mathbf{C}_i(y) \mathbf{C}_j(y)$ products modulo $F(y)$ yields a system of n linear equations over \mathbb{F}_q in the variables

$$\{v_i : 1 \leq i \leq n\} \cup \{w_{ij} : 1 \leq i < j \leq n\} \cup \{u_{3,k} : 1 \leq k \leq n\}.$$

To ease notation, we write $N = \frac{1}{2}n(n+3)$ for the number of variables, we write $\mathbf{t} = (v_i, w_{ij}, u_{3,k})$ for the vector consisting of these variables, and we let M be the $N \times n$ matrix giving the system of linear equations over \mathbb{F}_q . Then we obtain a matrix formula

$$(\mathbf{t} \ *) \begin{pmatrix} I_N & M \\ 0 & qI_n \end{pmatrix} = (\mathbf{t} \ \mathbf{0}).$$

This gives a lattice problem of dimension $N + n$ and determinant q^n with target vector \mathbf{t} .⁴

The size of the target \mathbf{t} depends on the distribution of the entries in U . For simplicity, suppose that the entries of U are chosen uniformly and independently from $\{-1, 0, 1\}$. Then each w_{ij} satisfies

$$\text{Prob}(w_{ij} = \pm 2) = \frac{2}{81}, \quad \text{Prob}(w_{ij} = \pm 1) = \frac{20}{81}, \quad \text{Prob}(w_{ij} = 0) = \frac{37}{81}.$$

So even ignoring the v_i and $u_{3,k}$ coordinates of \mathbf{t} , we find that the expected value of $\|\mathbf{t}\|_2^2$ is at least

$$\frac{n(n-1)}{2} \left(\frac{2}{81} \cdot 4 + \frac{20}{81} \cdot 1 + \frac{37}{81} \cdot 0 \right) = \frac{14}{81}(n^2 - n),$$

so we expect $\|\mathbf{t}\|$ to be roughly $n\sqrt{14}/9$. On the other hand, the Gaussian expected length of the shortest non-zero vector in the lattice is roughly

$$\sqrt{n^2/\pi e} \cdot (q^n)^{1/n^2} \approx 0.1 \cdot n \cdot q^{1/n}.$$

Thus even for quite large values of q , the target vector is likely to be considerably larger than many other (useless) short vectors in the lattice. For example, if $n = 100$, then the target only becomes a likely shortest vector if $q > 10^{111}$.

D A Non-Linear Attack

It is possible to use multiplication and reduction modulo $\mathbf{F}(y)$ in \mathbb{Y} to set up an attack in which one has to find small solutions to certain non-linear equations. Such problems appear to be completely infeasible, which we illustrate with a toy example with $n = 3$.

The attacker knows the polynomials

$$\mathbf{c}'(y) = c'_0 + c'_1 y + c'_2 y^2, \quad \mathbf{c}''(y) = c''_0 + c''_1 y + c''_2 y^2, \quad \mathbf{h}(y) = y^2 + h_0 y + h_1.$$

To make life easier, we take $\mathbf{h}(y) = y^3 + y + 1$. The attacker tries to find the small polynomials

$$\mathbf{m}'(x) = m'_0 + m'_1 x + m'_2 x^2 \quad \text{and} \quad \mathbf{m}''(x) = m''_0 + m''_1 x + m''_2 x^2$$

by eliminating the polynomial $\phi(y) = \phi_0 + \phi_1 y + \phi_2 y^2$ from the congruences

$$\begin{aligned} c'_0 + c'_1 y + c'_2 y^2 &\equiv m'_0 + m'_1(\phi_0 + \phi_1 y + \phi_2 y^2) + m'_2(\phi_0 + \phi_1 y + \phi_2 y^2)^2 \\ &\pmod{y^3 + y + 1}, \\ c''_0 + c''_1 y + c''_2 y^2 &\equiv m''_0 + m''_1(\phi_0 + \phi_1 y + \phi_2 y^2) + m''_2(\phi_0 + \phi_1 y + \phi_2 y^2)^2 \\ &\pmod{y^3 + y + 1}. \end{aligned}$$

⁴ With a bit more work, one can eliminate the $\mathbf{0}$ in the target and obtain a lattice problem of dimension N , but since N is so much larger than n , the gain is negligible.

Expanding and reducing modulo $y^3 + y + 1$, we find that

$$\begin{aligned} c'_0 + c'_1 y + c'_2 y^2 &= (m'_2 \phi_0^2 + m'_1 \phi_0 - 2m'_2 \phi_2 \phi_1 + m'_0) \\ &\quad + (2m'_2 \phi_1 \phi_0 - 2m'_2 \phi_2 \phi_1 + m'_1 \phi_1 - m'_2 \phi_2^2) y \\ &\quad + (2m'_2 \phi_2 \phi_0 + m'_2 \phi_1^2 - m'_2 \phi_2^2 + m'_1 \phi_2) y^2, \end{aligned}$$

and similarly for c'' . So we get 6 equations

$$\begin{aligned} m'_2 \phi_0^2 + m'_1 \phi_0 - 2m'_2 \phi_2 \phi_1 + m'_0 &= c'_0 \\ 2m'_2 \phi_1 \phi_0 - 2m'_2 \phi_2 \phi_1 + m'_1 \phi_1 - m'_2 \phi_2^2 &= c'_1 \\ 2m'_2 \phi_2 \phi_0 + m'_2 \phi_1^2 - m'_2 \phi_2^2 + m'_1 \phi_2 &= c'_2 \\ m''_2 \phi_0^2 + m''_1 \phi_0 - 2m''_2 \phi_2 \phi_1 + m''_0 &= c''_0 \\ 2m''_2 \phi_1 \phi_0 - 2m''_2 \phi_2 \phi_1 + m''_1 \phi_1 - m''_2 \phi_2^2 &= c''_1 \\ 2m''_2 \phi_2 \phi_0 + m''_2 \phi_1^2 - m''_2 \phi_2^2 + m''_1 \phi_2 &= c''_2 \end{aligned}$$

in the 9 variables $m'_0, m'_1, m'_2, m''_0, m''_1, m''_2, \phi_0, \phi_1, \phi_2$. These equations are linear in the small variables m'_i and m''_i , but are non-linear in the large variables ϕ_i that need to be eliminated. Eliminating the large variables, we are left with three highly non-linear polynomials in the six unknowns m'_i, m''_i . In other words, we need to find points with small coordinates on a 3-dimensional variety sitting in 6-dimensional space.

To investigate further, we computed an explicit example. We worked over \mathbb{F}_{11} and took $(c'_0, c'_1, c'_2, c''_0, c''_1, c''_2) = (1, 2, 3, 4, 5, 6)$. We used the Grobner-basis routine in Magma [1] to eliminate ϕ_0, ϕ_1, ϕ_2 from the 6 equations. The resulting equations for the 6 variables m'_i, m''_i covered more than two pages of small type and had no discernable structure.